When Is a Container a Comonad?

Danel Ahman
(University of Cambridge)

joint work with James Chapman and Tarmo Uustalu

Nottingham, 2 December 2011

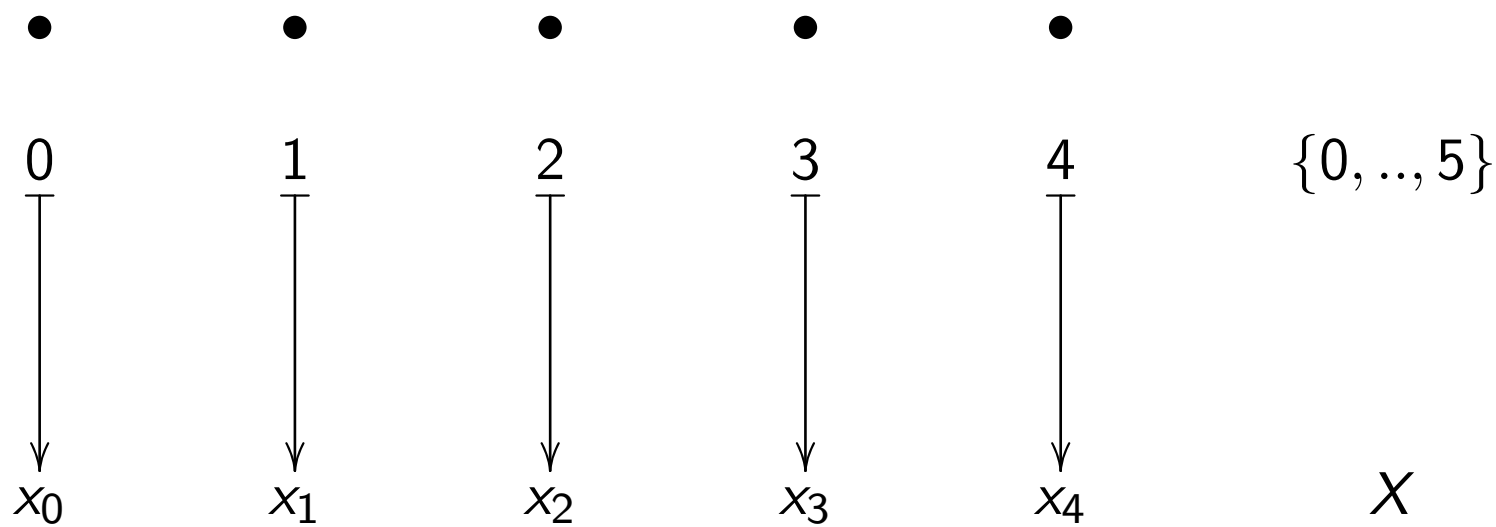
A motivating example

A motivating example

- How could one represent various datatypes generically?
- By intuition:
  - Many datatypes consist of generic "templates"
  - The actual data payload is mapped into this "template"

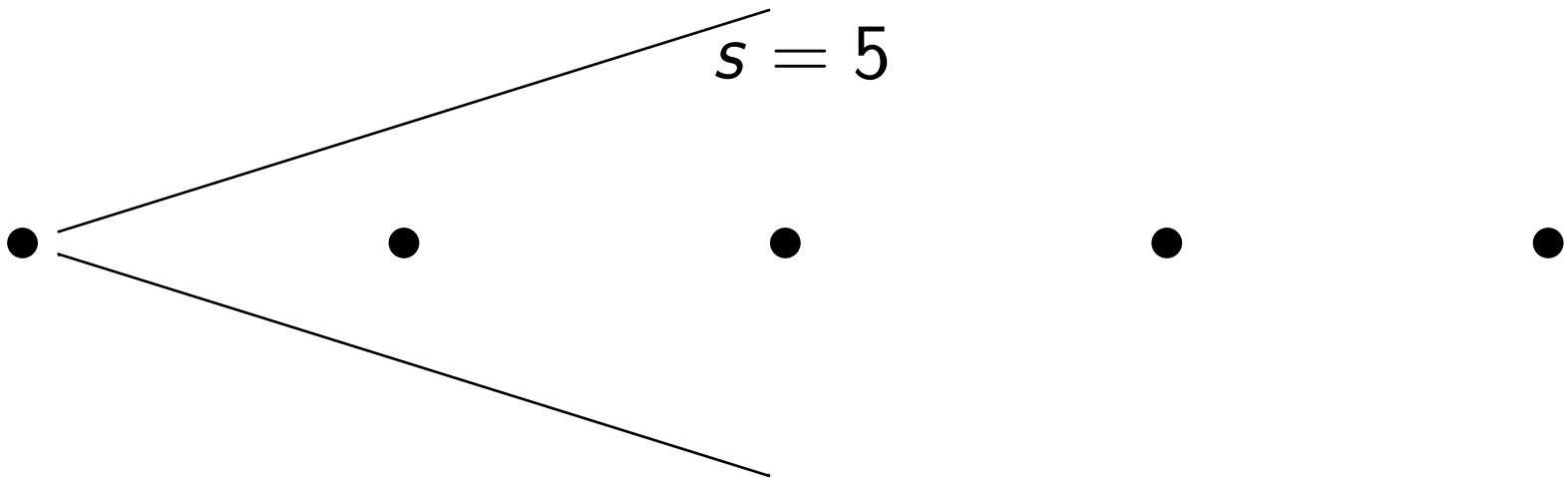
A motivating example

- How could one represent various datatypes generically?
- By intuition:
  - Many datatypes consist of generic "templates"
  - The actual data payload is mapped into this "template"



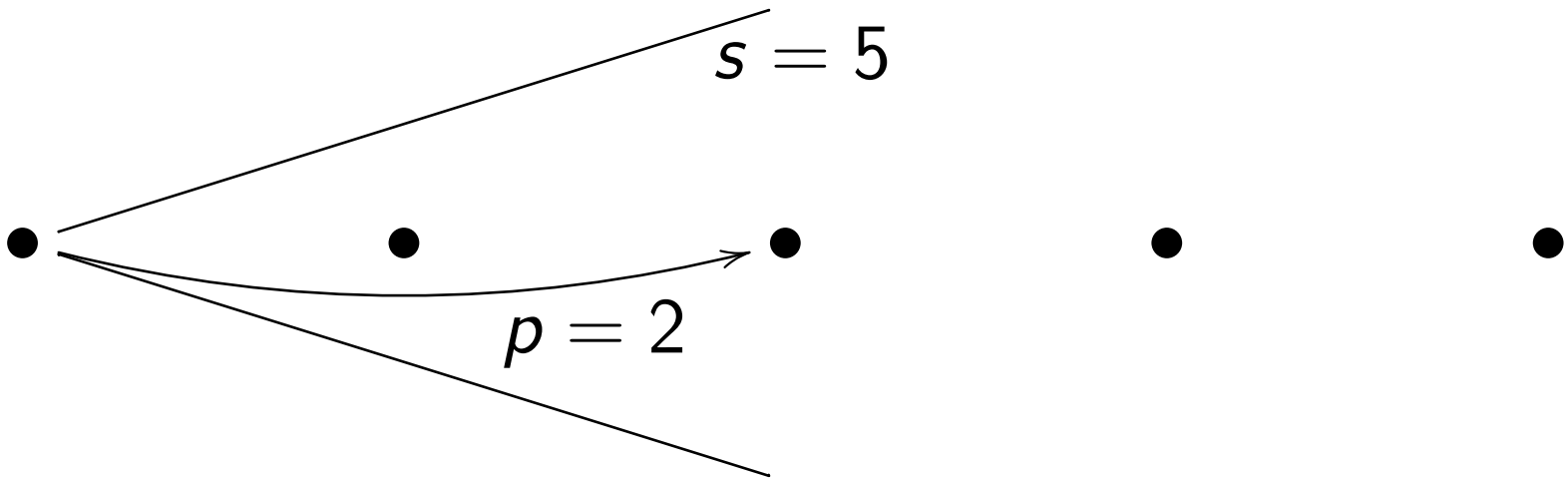
A motivating example ctd.

- Example: Lists
- Subshapes and additional structure?



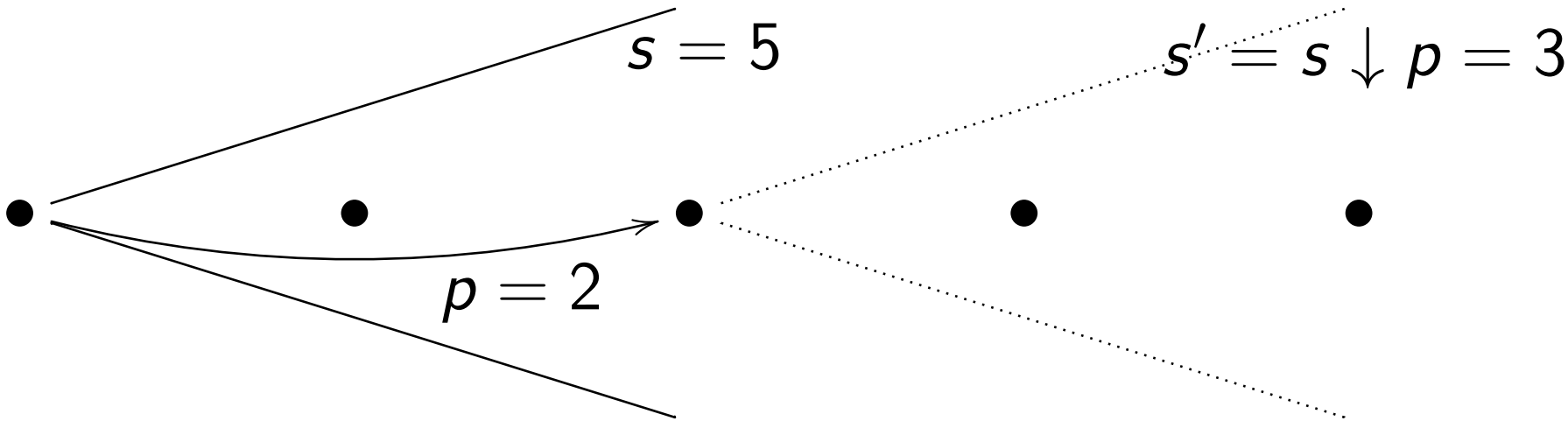
A motivating example ctd.

- Example: Lists
- Subshapes and additional structure?



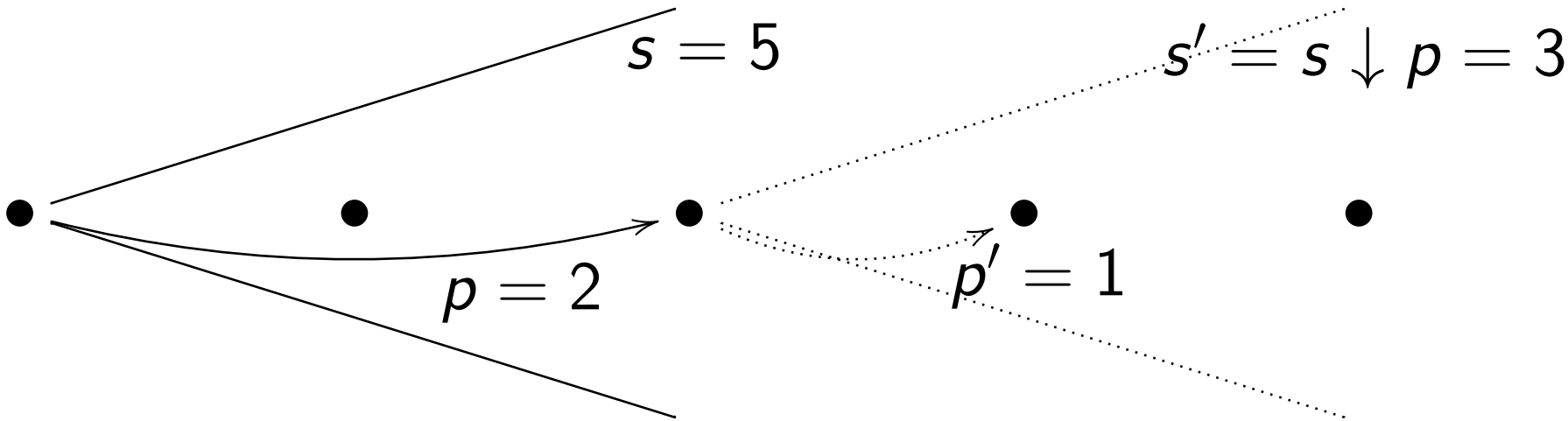
A motivating example ctd.

- Example: Lists
- Subshapes and additional structure?



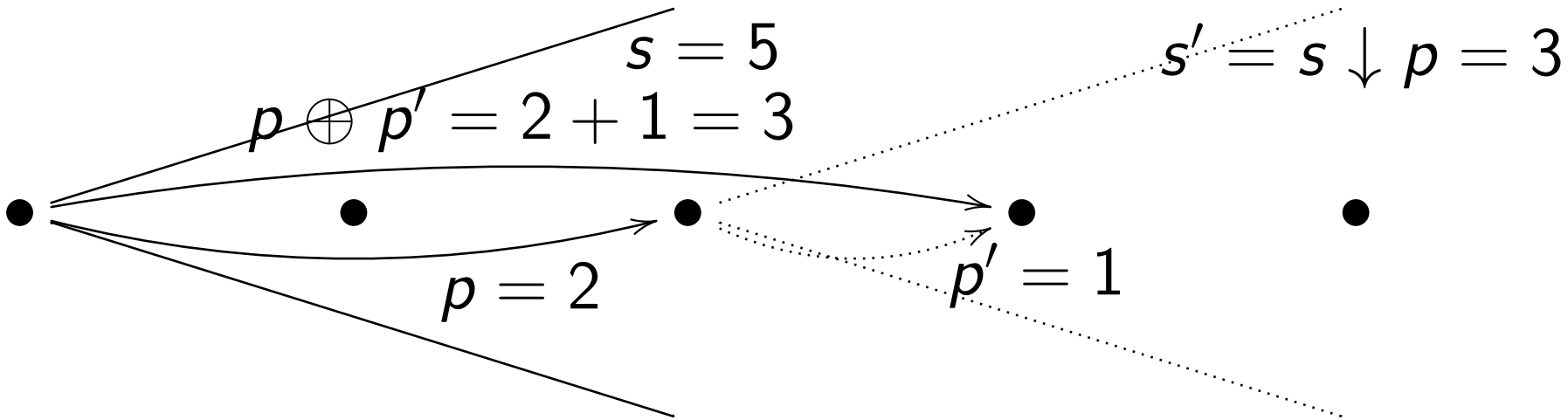
A motivating example ctd.

- Example: Lists
- Subshapes and additional structure?



A motivating example ctd.

- Example: Lists
- Subshapes and additional structure?



Containers

- Containers provide an elegant framework for analysing a large class of type constructors, in terms of shapes and positions in shapes

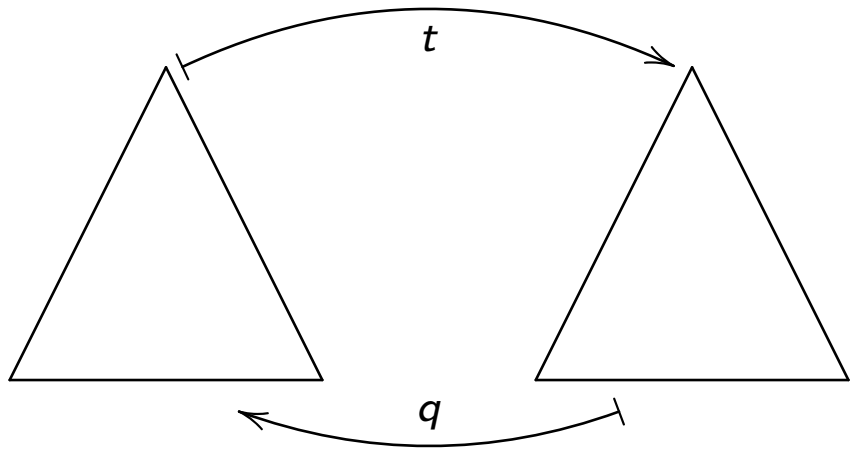
Directing containers?

- It often happens that there is a natural notion of a subshape given by a position in a shape
- Can we axiomatize this?
- What can we say about this class of type constructors?

Small recap on containers

Containers

- A container is given by
  - $S : \mathrm{Set}$ (*shapes*),
  - $P : S \to \mathrm{Set}$ (*positions*).
- A morphism between containers $S \lhd P$ and $S' \lhd P'$ is
  - $(t : S \to S' \quad \lhd \quad q : \Pi\{s : S\}.P'\,(t\,s) \to P\,s)$



- The identity morphism on the container $S \lhd P$ is
  - $\mathrm{id}^{\mathrm{c}} = (\mathrm{id}\,\{S\} \lhd \lambda\{s\}.\,\mathrm{id}\,\{P\,s\})$
- The composition of container morphs. $t \lhd q$ and $t' \lhd q'$
  - $(t' \lhd q') \circ^{\mathrm{c}} (t \lhd q) = ((t' \circ t) \lhd (\lambda\{s\}.\,q\,\{s\} \circ q'\,\{t\,s\}))$
- Containers form a category

Containers as functors

- Any container $S \lhd P$ defines a set functor $[\![S \lhd P]\!]^{\mathrm{c}}$ (its *interpretation*) by
  - $[\![S \lhd P]\!]^{\mathrm{c}} : \mathrm{Set} \to \mathrm{Set}$,
    $[\![S \lhd P]\!]^{\mathrm{c}}\, X = \Sigma s : S.\, P\, s \to X$
  - $[\![S \lhd P]\!]^{\mathrm{c}} : \forall \{X\}\, \{Y\}.\, (X \to Y) \to (\Sigma s : S.\, P\, s \to X) \to \Sigma s : S.\, P\, s \to Y$
    $[\![S \lhd P]\!]^{\mathrm{c}}\, f\, (s, v) = (s, f \circ v)$
- Any container morphism $t \lhd q$ between $S \lhd P$, $S' \lhd P'$ defines a natural transformation $[\![t \lhd q]\!]^{\mathrm{c}}$ between $[\![S \lhd P]\!]^{\mathrm{c}}$, $[\![S' \lhd P']\!]^{\mathrm{c}}$ by
  - $[\![t \lhd q]\!]^{\mathrm{c}} : \forall \{X\}.\, (\Sigma s : S.\, P\, s \to X) \to \Sigma s' : S'.\, P'\, s' \to X$
    $[\![t \lhd q]\!]^{\mathrm{c}}\, (s, v) = (t\, s, v \circ q\, \{s\})$
- $[\![-]\!]^{\mathrm{c}}$ is a functor from the category of containers to that of set functors.

Recap on containers

- The identity container $\mathrm{Id}^{\mathrm{c}} = (1 \lhd (\lambda *. 1))$
- The composition $C_0 \cdot^{\mathrm{c}} C_1$ of containers $C_0 = S_0 \lhd P_0$ and $C_1 = S_1 \lhd P_1$ is a container $S \lhd P$ where
  - $S = \Sigma s : S_0. P_0\, s \to S_1$
  - $P\,(s, v) = \Sigma p_0 : P_0\, s. P_1\,(v\, p_0)$
- The composition $h_0 \cdot^{\mathrm{c}} h_1$ of morphisms $h_0 = t_0 \lhd q_0$ and $h_1 = t_1 \lhd q_1$ is a container morphism $t \lhd q$ where
  - $t\,(s, v) = (t_0\, s, t_1 \circ v \circ q_0\,\{s\})$
  - $q\,\{s, v\}\,(p_0, p_1) = (q_0\,\{s\}\, p_0, q_1\,\{v\,(q_0\,\{s\}\, p_0)\}\, p_1)$
- Composition is a bifunctor

Recap on containers

- There are container isomorphisms
  - $\rho\,\{C\} : C \cdot^{\mathrm{c}} \mathsf{Id}^{\mathrm{c}} \to C$
  - $\lambda\,\{C\} : \mathsf{Id}^{\mathrm{c}} \cdot^{\mathrm{c}} C \to C$
  - $\alpha\,\{C_0\}\,\{C_1\},\{C_2\} : (C_0 \cdot^{\mathrm{c}} C_1) \cdot^{\mathrm{c}} C_2 \to C_0 \cdot^{\mathrm{c}} (C_1 \cdot^{\mathrm{c}} C_2)$

  satisfying appropriate coherence conditions

- Hence the category of containers is a monoidal category

- There are natural isomorphisms
  - $\mathsf{e} : \mathsf{Id} \to [\![\mathsf{Id}^{\mathrm{c}}]\!]^{\mathrm{c}}$
  - $\mathsf{m}\,\{C_0\}\,\{C_1\} : [\![C_0]\!]^{\mathrm{c}} \cdot [\![C_1]\!]^{\mathrm{c}} \to [\![C_0 \cdot^{\mathrm{c}} C_1]\!]^{\mathrm{c}}$

  satisfying appropriate coherence conditions.

- Hence interpretation $[\![-]\!]^{\mathrm{c}}$ is a monoidal functor.

Directed containers

$=$

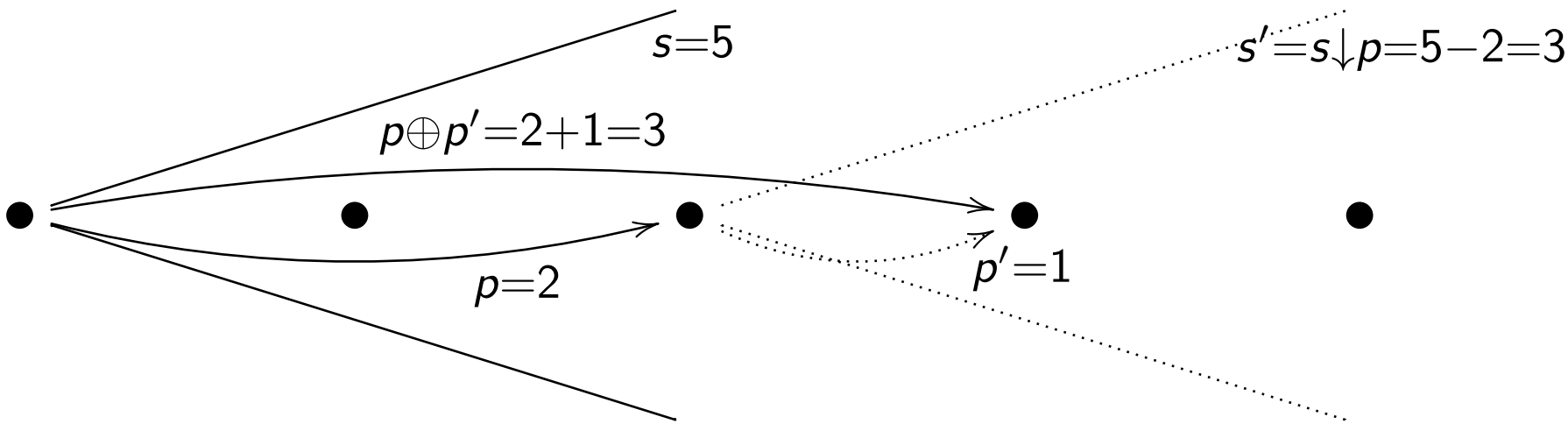
containers $+$ additional structure

Directed containers

- A directed container is given by a container $S \lhd P$
  - $S : \mathrm{Set}$ (*shapes*),
  - $P : S \to \mathrm{Set}$ (*positions*).

  and

  - $\downarrow : \Pi s : S.\, P\, s \to S$ (subshape),
  - $\mathsf{o} : \Pi\{s : S\}.\, P\, s$ (root position),
  - $\oplus : \Pi\{s : S\}.\, \Pi p : P\, s.\, P\,(s \downarrow p) \to P\, s$ (subshape positions)



Directed containers

- A directed container is given by a container $S \lhd P$
  - $S : \mathrm{Set}$ (*shapes*),
  - $P : S \to \mathrm{Set}$ (*positions*).

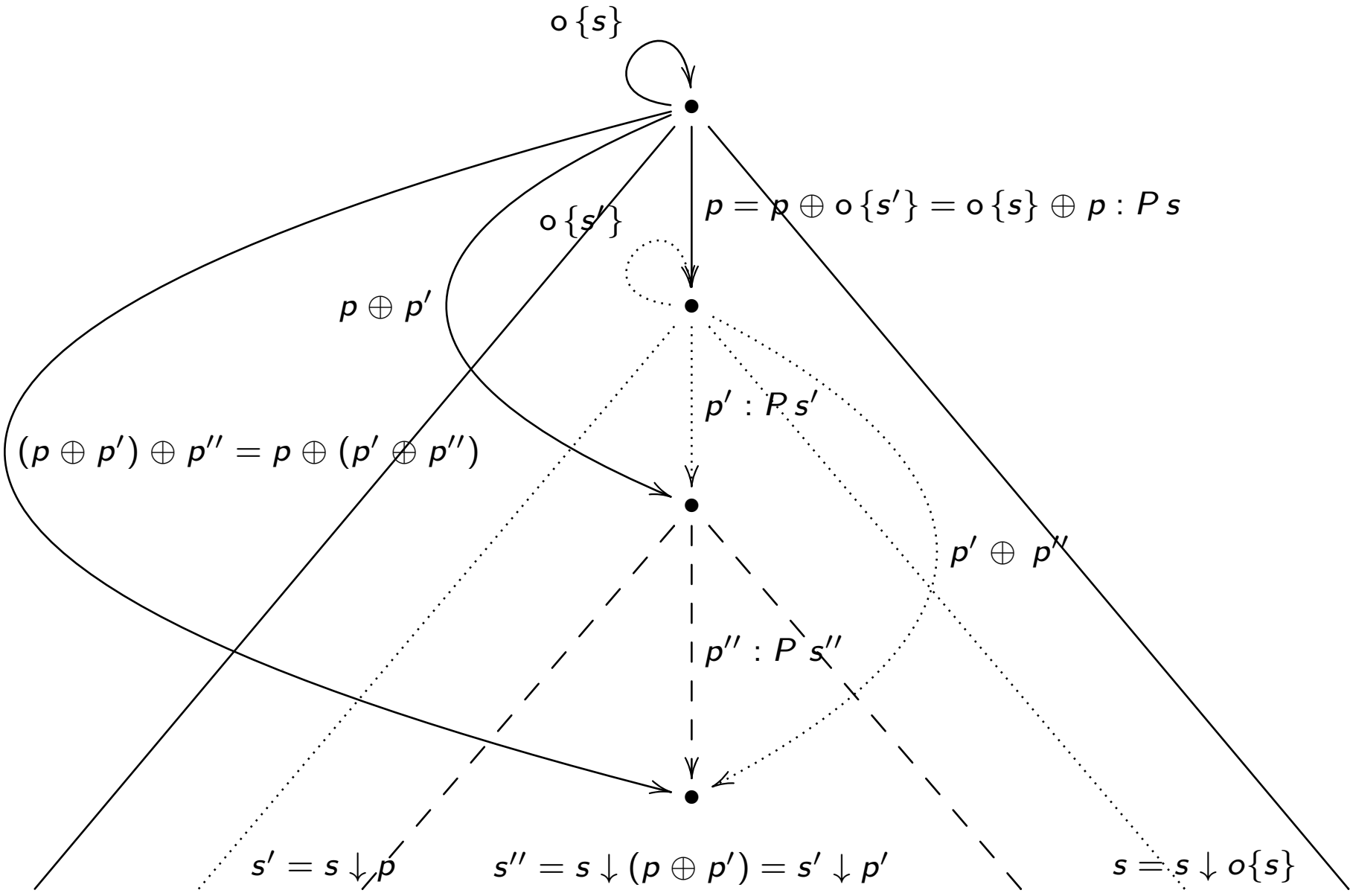
  and

  - $\downarrow : \Pi s : S.\, P\, s \to S$ (subshape),
  - $\mathsf{o} : \Pi\{s : S\}.\, P\, s$ (root position),
  - $\oplus : \Pi\{s : S\}.\, \Pi p : P\, s.\, P\,(s \downarrow p) \to P\, s$ (subshape positions)

  such that

  - $\forall\{s\}.\, s \downarrow \mathsf{o} = s$,
  - $\forall\{s, p, p'\}.\, s \downarrow (p \oplus p') = (s \downarrow p) \downarrow p'$,
  - $\forall\{s, p\}.\, p \oplus\{s\} \mathsf{o} = p$,
  - $\forall\{s, p\}.\, \mathsf{o}\{s\} \oplus p = p$,
  - $\forall\{s, p, p', p''\}.\, (p \oplus\{s\} p') \oplus p'' = p \oplus (p' \oplus p'')$.

Directed containers ctd.



Directed containers morphisms

- A directed container morphism between $(S \lhd P, \downarrow, \mathsf{o}, \oplus)$ and $(S' \lhd P', \downarrow', \mathsf{o}', \oplus')$ is a container morphism $t \lhd q$
  - $t : S \to S' \quad , \quad q : \Pi\{s : S\}.P'\,(t\,s) \to P\,s$

  such that
  - $\forall\{s, p\}.\, t\,(s \downarrow q\,p) = t\,s \downarrow' p,$
  - $\forall\{s\}.\, \mathsf{o}\,\{s\} = q\,(\mathsf{o}'\,\{t\,s\}),$
  - $\forall\{s, p, p'\}.\, q\,p \oplus \{s\}\,q\,p' = q\,(p \oplus' \{t\,s\}\,p')$

Directed containers morphisms

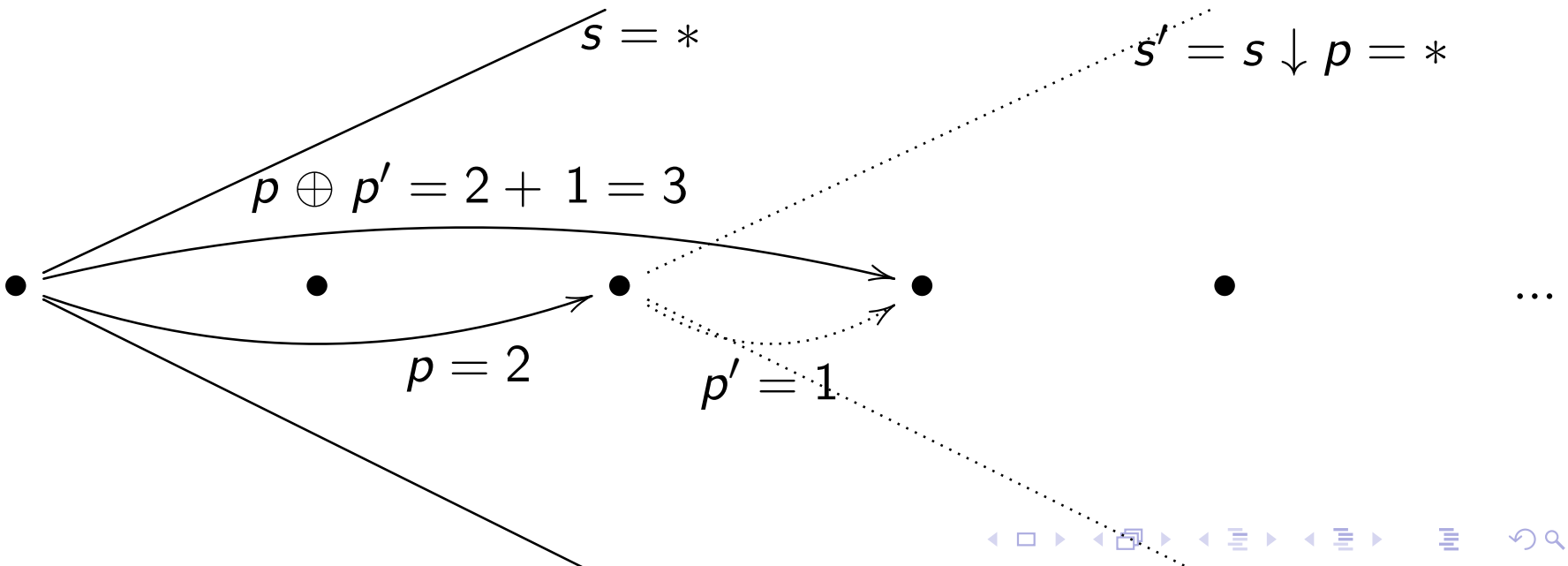
- A directed container morphism between $(S \lhd P, \downarrow, \mathrm{o}, \oplus)$ and $(S' \lhd P', \downarrow', \mathrm{o}', \oplus')$ is a container morphism $t \lhd q$
  - $t : S \to S' \quad , \quad q : \Pi\{s : S\}.P'\,(t\,s) \to P\,s$

  such that
  - $\forall\{s, p\}.\, t\,(s \downarrow q\,p) = t\,s \downarrow' p,$
  - $\forall\{s\}.\, \mathrm{o}\,\{s\} = q\,(\mathrm{o}'\,\{t\,s\}),$
  - $\forall\{s, p, p'\}.\, q\,p \oplus \{s\}\,q\,p' = q\,(p \oplus' \{t\,s\}\,p')$
- With identities and composition
  - $\mathrm{id}^{\mathrm{dc}} = \mathrm{id}\,\{S\} \lhd (\lambda\{s\}.\,\mathrm{id}\,\{P\,s\})$
  - $(t' \lhd q') \circ^{\mathrm{dc}} (t \lhd q) = (t' \circ t) \lhd (\lambda\{s\}.\, q\,\{s\} \circ q'\,\{t\,s\})$
- Hence directed containers form a category

Streams - Lists - Trees

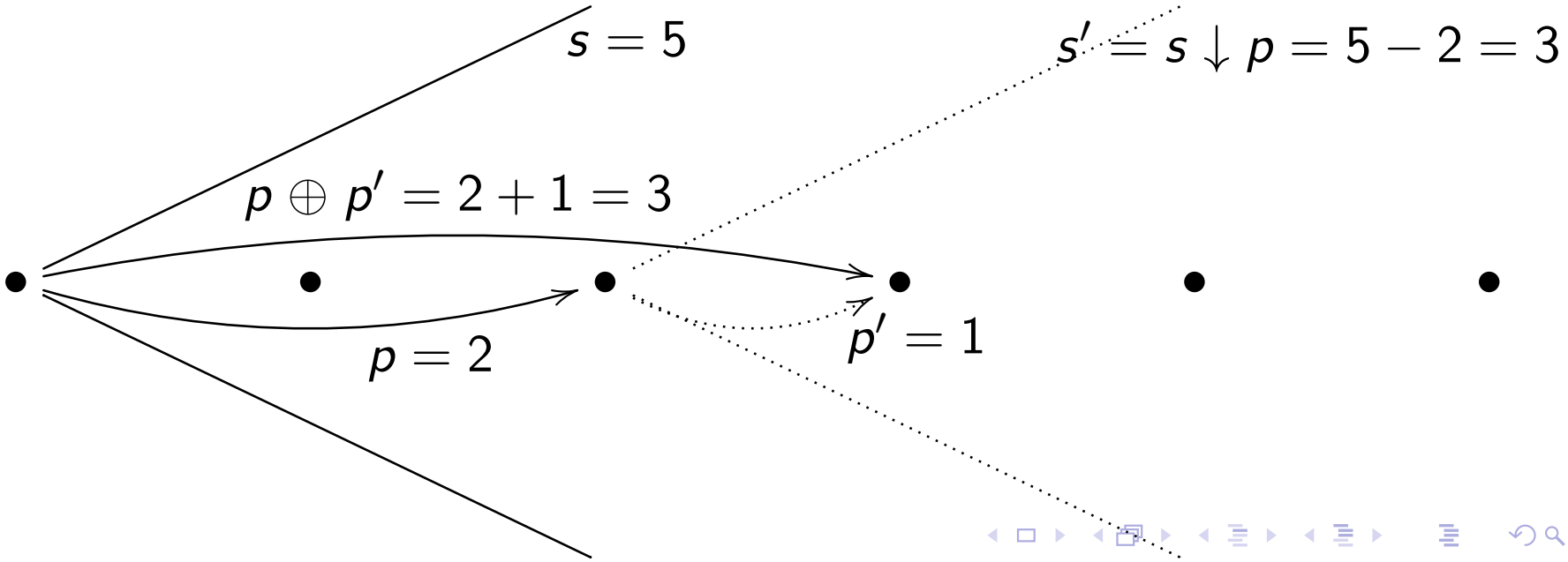
Examples of directed containers

- **Streams**
  - $S = 1$ *(shapes)*
  - $P* = \mathsf{Nat}$ *(positions)*
  - $s \downarrow p = s$
  - $\mathsf{o} = \mathit{zero}$
  - $p \oplus p' = p + p'$



Examples of directed containers ctd.

- **Non-empty lists**
  - $S = \mathsf{Nat}$ *(shapes)*
  - $P\,s = \mathsf{Fin}\,(\mathsf{suc}\,s)$ *(positions)*
  - $s \downarrow p = s - p$
  - $\mathsf{o} = \mathit{zero}$
  - $p \oplus \{s\}\, p' = p + p'$



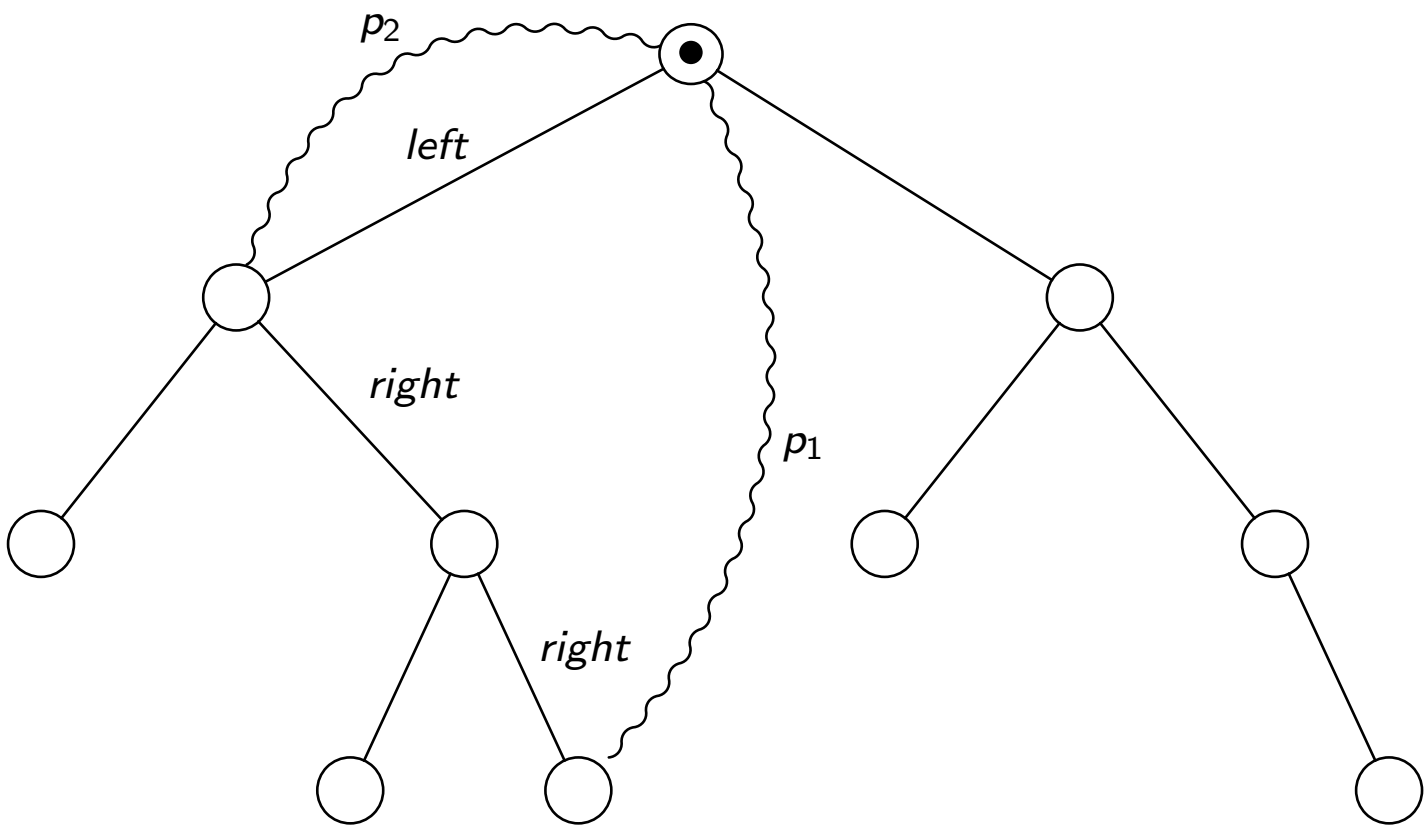
Examples of directed containers ctd.

- **Node-labeled binary trees**
  - data TShp : Set where
    nd : Maybe TShp → Maybe TShp → TShp
  - data TPos : TShp → Set where
    stop : ∀{s}→ TPos s
    left : ∀{lr}→ TPos l → TPos (nd (just l) r)
    right : ∀{lr}→ TPos r → TPos (nd l (just r))

Examples of directed containers ctd.

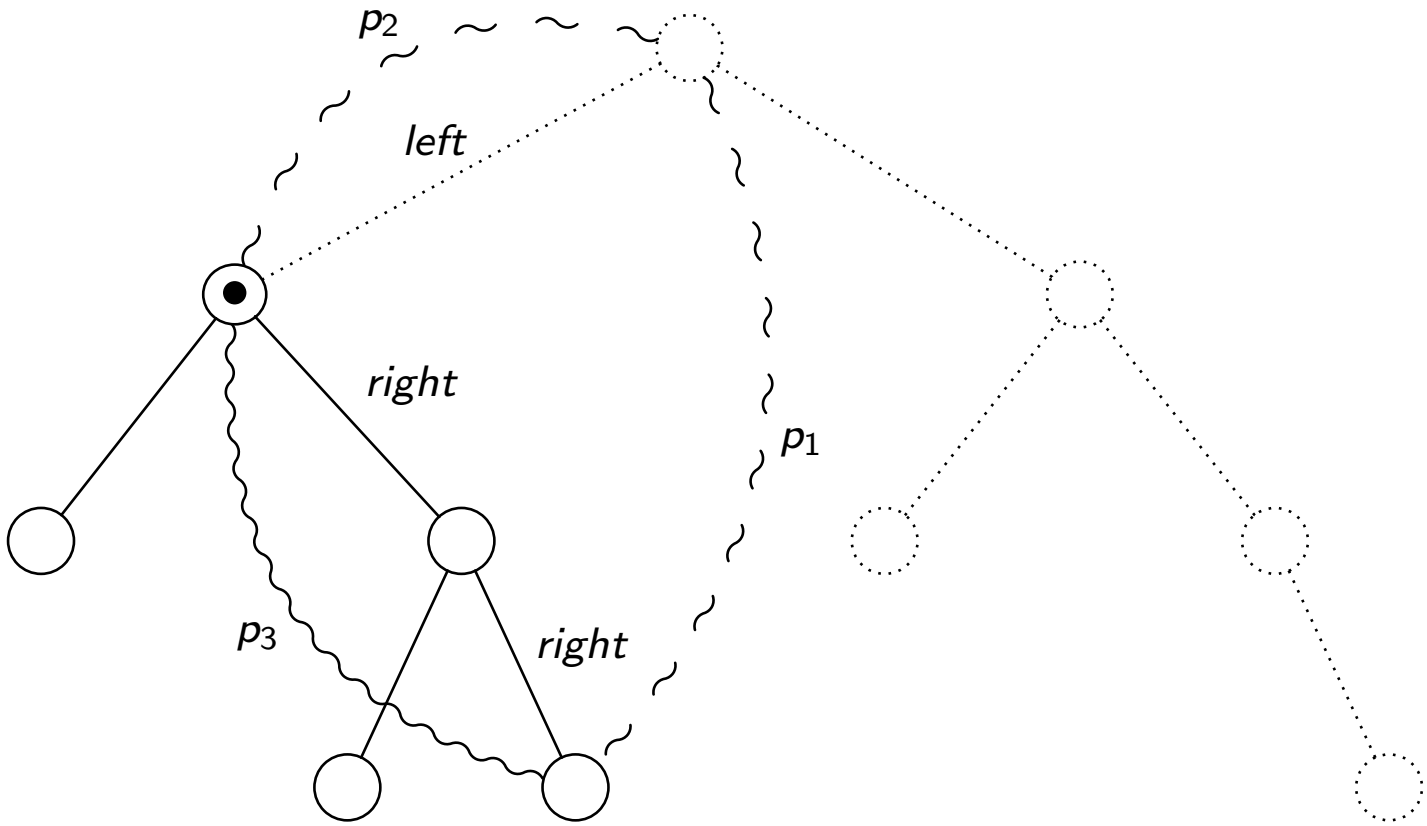
- **Node-labeled binary trees**
  - data TShp : Set where
    nd : Maybe TShp → Maybe TShp → TShp
  - data TPos : TShp → Set where
    stop : ∀{s}→ TPos s
    left : ∀{lr}→ TPos l → TPos (nd (just l) r)
    right : ∀{lr}→ TPos r → TPos (nd l (just r))
- **Node-labeled binary trees as a directed container**
  - $S = TShp$, $P\,s = TPos\,s$
  - $s \downarrow stop = s$
    $(nd\,(just l)\,r) \downarrow (left\,p) = l \downarrow p$
    $(nd\,l\,(just\,r)) \downarrow (right\,p) = r \downarrow p$
  - o $= stop$
  - $p \oplus p' =$ **concatenate**(p, p')

Examples of directed containers ctd.



- $p_1 = \mathit{left}(\mathit{right}(\mathit{right}\ \mathit{stop})) : \mathsf{TShp}\, s$
- $p_2 = \mathit{left}\ \mathit{stop} : \mathsf{TShp}\, s$

Examples of directed containers ctd.



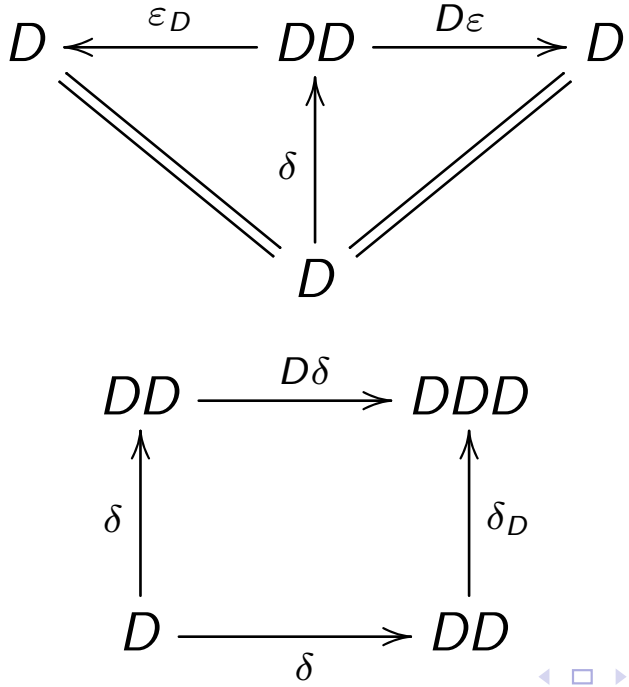
- $p_2 = \mathit{left}\ \mathit{stop} : \mathsf{TShp}\, s$
- $p_3 = \mathit{right}(\mathit{right}\ \mathit{stop}) : \mathsf{TShp}\,(s \downarrow p_2)$
- $p_1 = \mathit{left}(\mathit{right}(\mathit{right}\ \mathit{stop})) = p_2 \oplus p_3 : \mathsf{TShp}\, s$

Comonads

Comonads

- A comonad on a category $\mathbb{C}$ is
  - $D : \mathbb{C} \to \mathbb{C}$ *(underlying functor)*
  - $\varepsilon : D \to \mathsf{Id}$ *(counit)*
  - $\delta : D \to DD$ *(comultiplication)*

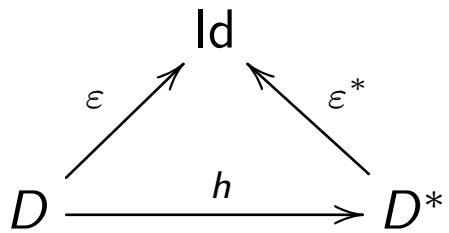
  satisfying the following laws

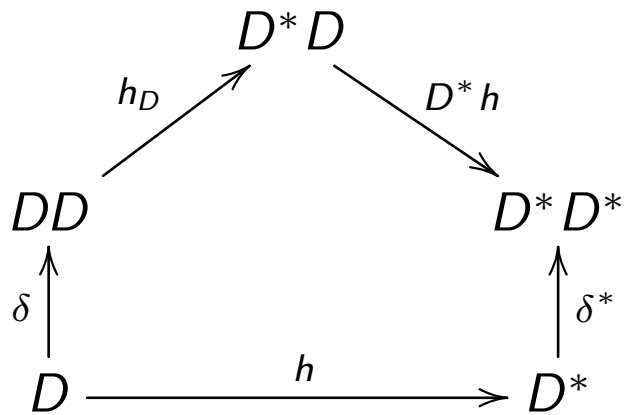


Comonads morphisms

- A comonad morphism between comonads (D, ε, δ) and $(D^*, \varepsilon^*, \delta^*)$ is a natural transformation
  - $h : D \to D^*$

  satisfying the following laws





Directed containers as comonads

Directed containers as comonads

- Any directed container $E = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$ defines a comonad $[\![E]\!]^{\mathrm{dc}} = (D, \varepsilon, \delta)$ where

  - $D = [\![S \lhd P]\!]^{\mathrm{c}}$

    $D\,X = \Sigma s : S.\, P\,s \to X$

    $D\,f\,(s, v) = (s, f \circ v)$

  - $\varepsilon : \forall\{X\}.(\Sigma s : S.\, P\,s \to X) \to X$

    $\varepsilon\,(s, v) = v\,(\mathsf{o}\,\{s\})$

  - $\delta : \forall\{X\}.\,(\Sigma s : S.\, P\,s \to X) \to$
    $\Sigma s : S.\, P\,s \to \Sigma s' : S.\, P\,s' \to X$

    $\delta\,(s, v) = (s, \lambda p.\,(s \downarrow p, \lambda p'.\, v\,(p \oplus \{s\}\,p'))).$

Directed container morphisms as comonad morphisms

- Any morphism $t \lhd q$ between directed containers $E = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$ and $E' = (S' \lhd P', \downarrow', \mathsf{o}', \oplus')$ defines a comonad morphism $[\![t \lhd q]\!]^{\mathrm{dc}} = [\![t \lhd q]\!]^{\mathrm{c}}$ between $[\![E]\!]^{\mathrm{dc}}$ and $[\![E']\!]^{\mathrm{dc}}$

- $[\![t \lhd q]\!]^{\mathrm{dc}} : \forall \{X\}.\, (\Sigma s : S.\, P\, s \to X) \to \Sigma s' : S'.\, P'\, s' \to X$
  $[\![t \lhd q]\!]^{\mathrm{dc}}\, (s, v) = (t\, s, v \circ q\, \{s\})$

- $[\![-]\!]^{\mathrm{dc}}$ preserves the identities and composition

- $[\![-]\!]^{\mathrm{dc}}$ is a functor

Comonad morphisms as directed container morphisms

- For $E = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$ and $E' = (S' \lhd P', \downarrow', \mathsf{o}', \oplus')$, a comonad morphism τ between $[\![E]\!]^{\mathrm{dc}}$ and $[\![E']\!]^{\mathrm{dc}}$ defines directed container morphism $\ulcorner \tau \urcorner^{\mathrm{dc}}$ between E, E'
  - $t : S \to S'$
    $t\, s = \mathsf{fst}\,(\tau\,\{P\, s\}\,(s, \mathsf{id}))$
  - $q : \Pi\{s : S\}.\, P'\,(t\, s) \to P\, s$
    $q\,\{s\} = \mathsf{snd}\,(\tau\,\{P\, s\}\,(s, \mathsf{id}))$
- Moreover:
  - $\ulcorner [\![h]\!]^{\mathrm{dc}} \urcorner^{\mathrm{dc}} = h$
  - $\ulcorner \tau \urcorner^{\mathrm{dc}} = \ulcorner \tau' \urcorner^{\mathrm{dc}}$ implies $\tau = \tau'$
- Hence $[\![-]\!]^{\mathrm{dc}}$ is fully faithful

Containers $\cap$ comonads

$=$

directed containers

Containers ∩ comonads = directed containers

- A comonad $(\llbracket S \lhd P \rrbracket^{\mathrm{c}}, \varepsilon, \delta)$ determines a directed container $\lceil (\llbracket S \lhd P \rrbracket^{\mathrm{c}}, \varepsilon, \delta) \rceil = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$
  where
  - $s \downarrow p = \mathsf{snd}\,(t^{\delta}\, s)\, p$
  - $\mathsf{o}\,\{s\} = q^{\varepsilon}\{s\} *$
  - $p \oplus \{s\}\, p' = q^{\delta}\,\{s\}\,(p, p')$

  using the container morphisms
  - $t^{\varepsilon} \lhd q^{\varepsilon} : S \lhd P \to \mathsf{Id}^{\mathrm{c}}$
    $t^{\varepsilon} \lhd q^{\varepsilon} = \ulcorner \mathsf{e} \circ \varepsilon \urcorner^{\mathrm{c}}$
  - $t^{\delta} \lhd q^{\delta} : S \lhd P \to (S \lhd P) \cdot^{\mathrm{c}} (S \lhd P)$
    $t^{\delta} \lhd q^{\delta} = \ulcorner \mathsf{m}\,\{S \lhd P\}\,\{S \lhd P\} \circ \delta \urcorner^{\mathrm{c}}$
- It is also true that
  - $\forall \{s\}.\, t^{\varepsilon}\, s = *$ and $\forall \{s\}.\, \mathsf{fst}\,(t^{\delta}\, s) = s$

Containers $\cap$ comonads $=$ directed containers ctd.

- We have two lemmas
  - $[\![\lceil([\![S \lhd P]\!]^{\mathrm{c}}, \varepsilon, \delta)\rceil]\!]^{\mathrm{dc}} = ([\![S \lhd P]\!]^{\mathrm{c}}, \varepsilon, \delta)$
  - $\lceil[\![S \lhd P, \downarrow, \mathsf{o}, \oplus]\!]^{\mathrm{dc}}\rceil = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$

Containers ∩ comonads = directed containers ctd.

- We have two lemmas
  - $[\![\lceil([\![S \lhd P]\!]^{\mathrm{c}}, \varepsilon, \delta)\rceil]\!]^{\mathrm{dc}} = ([\![S \lhd P]\!]^{\mathrm{c}}, \varepsilon, \delta)$
  - $\lceil[\![S \lhd P, \downarrow, \mathsf{o}, \oplus]\!]^{\mathrm{dc}}\rceil = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$

- And we have a pullback in **Cat**:

$$\begin{array}{ccc} \mathbf{DCont} & \xrightarrow{U} & \mathbf{Cont} \\ {\scriptstyle [\![-]\!]^{\mathrm{dc}}}\downarrow{\scriptstyle \text{f.f.}} & & {\scriptstyle [\![-]\!]^{\mathrm{c}}}\downarrow{\scriptstyle \text{f.f.}} \\ \mathbf{Cmnd(Set)} & \xrightarrow{U} & \mathbf{[Set, Set]} \end{array}$$

Designated focus positions

Focussing a container

- Given a container $C_0 = (S_0 \lhd P_0)$, we can focus it by defining a directed container $E = (S \lhd P, \downarrow, \mathsf{o}, \oplus)$ where
  - $S = \Sigma s : S_0 . P_0\, s$
  - $P\,(s, p) = P_0\, s$
  - $\mathsf{o}\,\{s, p\} = p$
  - $(s, p) \downarrow p' = (s, p')$
  - $p' \oplus \{s, p\}\, p'' = p''$

Focussing a container

- Given a container $C_0 = (S_0 \lhd P_0)$, we can focus it by defining a directed container $E = (S \lhd P, \downarrow, \mathrm{o}, \oplus)$ where
  - $S = \Sigma s : S_0 . P_0\, s$
  - $P\,(s, p) = P_0\, s$
  - $\mathrm{o}\,\{s, p\} = p$
  - $(s, p) \downarrow p' = (s, p')$
  - $p' \oplus \{s, p\}\, p'' = p''$
- On binary trees:
  - $C_0 = (TShp \lhd TPos)$
  - $(s, p) : S$ - tree shape s and tree position p we focus on
  - $p : P$ - a tree position
  - o - the focussed tree position
  - $\downarrow$, $\oplus$ - focussing on another position

Zippers

- *Zippers* consist of a datatype with a focus (eg. tree, list) and a context (some path structure)

Zippers

- *Zippers* consist of a datatype with a focus (eg. tree, list) and a context (some path structure)
- A simple example: non-empty lists with focus
- $S = \mathsf{Nat} \times \mathsf{Nat}$ (shapes)
- $P\,(s_0, s_1) = [-s_0..s_1]$ (positions)

$$\begin{array}{rcl} [\![S, P]\!]^{\mathrm{c}}\, X & = & \Sigma s : S.\, P, s \to X \\ & = & \Sigma (s_0, s_1) : \mathsf{Nat} \times \mathsf{Nat}.\, [-s_0..s_1] \to X \\ & \cong & \mathsf{List}\, X \times X \times \mathsf{List}\, X \end{array}$$

Zippers

- *Zippers* consist of a datatype with a focus (eg. tree, list) and a context (some path structure)
- A simple example: non-empty lists with focus
- $S = \mathrm{Nat} \times \mathrm{Nat}$ (shapes)
- $P\,(s_0, s_1) = [-s_0..s_1]$ (positions)

We can define

- $(s_0, s_1) \downarrow p = (s_0 + p, s_1 - p)$ (subshape at a position)
- $\mathrm{o}\,\{s_0, s_1\} = 0$ (root position)
- $p \oplus \{s_0, s_1\}p' = p + p'$ (positions in the subshape)

Other constructions and further work

- Constructions not described in this talk:
  - tree zippers
  - strict directed containers
  - products and composition of (strict) directed containers
  - functions from a monoid to directed container
  - distributive laws between directed containers
  - cofree construction on directed containers
  - monad structure on containers

Containers $\cap$ Monads $=$?

- Given a container $C = (S \lhd P)$
- The structure (η, μ) of a monad on $[\![S \lhd P]\!]^{\mathrm{c}}$ could be represented as

  - $\mathsf{e} : S$ *(for the shape map for η)*
  - $\bullet : \Pi s : S.(P\, s \to S) \to S$ *(for the shape map for μ)*
  - $\nwarrow : \Pi\{s : S\}.\, \Pi v : P\, s \to S.P\,(s \bullet v) \to P\, s$
    (for the position map for μ)
  - $\nearrow : \Pi\{s : S\}.\, \Pi v : P\, s \to S.$
    $\Pi p : P\,(s \bullet v).\, P\,(v\,(v \nwarrow \{s\}p))$
    (for the position map for μ)

Conclusion

- Directed containers give us a theory to reason about datatypes carrying additional structure
- Directed containers also form a category
- Directed containers are the same as containers that are comonads
- Various examples: lists, streams, trees, zippers
- We have a formalisation in Agda
- http://cs.ioc.ee/~danel/dcont.html